CLAIMS

What is claimed is:

1. A microprocessor including:

an instruction decoder to decode multiple threads of instructions, the instruction decoder having,

an instruction decode pipeline to decode each input instruction associated with each thread; and

a shadow pipeline operating in parallel with the instruction decode pipeline, the shadow pipeline including,

an instruction thread identification pipeline to associate a thread identification with each instruction being decoded in the instruction decode pipeline, and

an instruction valid pipeline to associate a valid indicator with each instruction being decoded in the instruction decode pipeline.

2. The microprocessor of claim 1 wherein the instruction decode pipeline and shadow pipeline are physically integrated together into one pipeline within a microprocessor integrated circuit.

3. The microprocessor of claim 1 wherein the instruction decoder further has,

a pipeline controller coupled to the instruction decode pipeline and the shadow pipeline, the pipeline controller to control in parallel the clocking of each pipestage of the instruction decode pipeline and each pipestage of the shadow pipeline.

4. The microprocessor of claim 3 wherein a thread specific clear is received and the pipeline controller invalidates only those instructions in each stalled pipestage of the instruction decode pipeline having a thread identification the same as the thread identification of the thread specific clear.

5. The microprocessor of claim 4 wherein the pipeline controller invalidates instructions in each pipestage by setting or clearing the valid indicator in the parallel pipestages of the instruction valid pipeline.

6. The microprocessor of claim 3 wherein a thread specific clear is received and the pipeline controller invalidates only those instructions input into each clocked pipestage of

the instruction decode pipeline when a pipestage prior to each clocked pipestage has a thread identification the same as the thread identification of the thread specific clear.

7. The microprocessor of claim 6 wherein the pipeline controller invalidates instructions in each pipestage by setting or clearing the valid indicator in the parallel pipestages of the instruction valid pipeline.

8. The microprocessor of claim 3 wherein a stall associated with a thread identification is received and the pipeline controller determines that the thread identification associated with the stall matches the thread identification of the valid instruction in the next to last pipestage of the instruction decode pipeline and the pipeline controller stalls the next to last pipestage by stopping the clock to the next to last pipestage for the next cycle to hold the valid instruction contained therein.

9. The microprocessor of claim 8 wherein the pipeline controller further determines that a valid instruction is contained within a first pipestage of the instruction decode pipeline other than the next to last pipestage and a valid instruction is contained within a second pipestage

immediately subsequent to the first pipestage and the pipeline controller stalls the first pipestage by stopping the clock to the first pipestage for the next cycle to hold the valid instruction contained therein.

10. The microprocessor of claim 9 wherein the pipeline controller determines that an instruction is valid by analyzing the valid indicator within the instruction valid pipeline of each pipestage.

11. The microprocessor of claim 3 wherein the pipe controller further determines that an invalid instruction is contained within a first pipestage immediately prior to a second pipestage in the instruction decode pipeline and the pipe controller powers down the second pipestage by stopping the clock to the second pipestage until a valid instruction is contained within the first pipestage to conserve power.

12. A multithread pipelined instruction decoder comprising:

an instruction decode pipeline to decode instructions associated with a plurality of instruction threads; and a shadow pipeline operating in parallel with the instruction decode pipeline, the shadow pipeline including,

an instruction thread identification pipeline to associate a thread identification at each pipe stage with each instruction being decoded in the instruction decode pipeline, and

an instruction valid pipeline to associated a valid indicator at each pipe stage with each instruction being decoded in the instruction decode pipeline.

13. The multithread pipelined instruction decoder of claim 12 further comprising:

a pipeline controller coupled to the instruction decode pipeline and the shadow pipeline, the pipeline controller to control in parallel the clocking of each pipestage of the instruction decode pipeline and each pipestage of the shadow pipeline.

14. The multithread pipelined instruction decoder of claim 13 wherein the pipeline controller further controls the invalidation of instructions in each pipestage of the instruction decode pipeline by setting the valid indicator in

each pipestage of the instruction valid pipeline to indicate an invalid instruction.

15. The multithread pipelined instruction decoder of claim 13 wherein the pipeline controller includes powerdown logic to analyze the valid indicator of each pipestage to determine if the next pipestage can be powerdown and to determine if each pipestage should be stalled and the pipeline controller further including clock control logic to determine if a clock signal to each pipestage of the instruction decode pipeline and the shadow pipeline can be stopped to conserve power or preserve data during a stall.

16. The multithread pipelined instruction decoder of claim 15 wherein the powerdown logic of the pipeline controller to analyze the valid indicator of each pipestage to determine if the next pipestage can be powerdown includes,

an inverter for each pipestage to invert the valid indicator to determine if the next pipestage can be powerdown for a next clock cycle.

17. The multithread pipelined instruction decoder of claim 15 wherein the powerdown logic of the pipeline controller to

analyze the valid indicator of each pipestage to determine if each pipestage should be stalled includes,

for a determination of a stall condition for the next to last pipestage,

an XOR gate to exclusively OR the thread identification of the next to last pipestage with the thread identification of the stall to determined if they match, and

a first AND gate to AND the valid indicator of the next to last pipestage with the output from the XOR gate to determine if a pipestage before the next last pipestage should be stalled.

18. The multithread pipelined instruction decoder of claim 17 wherein the powerdown logic of the pipeline controller to analyze the valid indicator of each pipestage to determine if each pipestage should be stalled includes,

for a determination of a stall condition for any other pipestage but the next to last, for each pipestage including

a second AND gate to AND the valid indicator of the pipestage for which the determination is being made with the valid indicator of the next pipestage, and

a third AND gate to AND the output of the second AND gate with the output from the first AND gate to determine if a given pipestage other than the next to last pipestage should be stalled.

19. A method of decoding multiple threads of instructions, comprising:

inputting an instruction of a first thread of instructions, a first instruction thread identification, and a first instruction valid indicator into a pipeline in parallel;

decoding the instruction of the first thread of instructions;

maintaining the parallel association between the instruction of the first thread, the first instruction thread identification, and the first instruction valid indicator during the decoding of the instruction of the first thread of instructions;

inputting an instruction of a second thread of instructions, a second instruction thread identification, and a second instruction valid indicator into a pipeline in parallel;

decoding the instruction of the second thread of instructions; and

maintaining the parallel association between the instruction of the second thread of instructions, the second instruction thread identification, and the second instruction valid indicator during the decoding of the instruction of the second thread of instructions.

20. The method of claim 19 further comprising:

invalidating only those instructions having a first instruction thread identification in the pipeline when a thread specific clear for the first instruction thread is received.

21. The method of claim 20 wherein,

the instructions are invalidated by clearing the first instruction valid indicator for each instruction of the first thread of instructions in the pipeline.

22. The method of claim 19 further comprising:

disabling a clock signal to a next to last pipestage in the pipeline when the instruction valid indicator of the instruction contained within the next to last pipestage indicates a valid instruction and a thread identification of a thread specific stall matches the thread identification of the instruction contained within the next to last pipestage.

23. The method of claim 19 further comprising:

disabling a clock signal to a pipestage other than the next to last pipestage in the pipeline when the instruction valid indicator of the instruction contained within the pipestage being evaluated indicates a valid instruction and the instruction valid indicator of the instruction of a next pipestage indicates a valid instruction and the next to last pipestage is stalled.

24. The method of claim 19 further comprising:

enabling a clock signal to a pipestage when the pipestage is not stalled and the pipestage is not powerdown.

25. The method of claim 19 further comprising:

disabling a clock signal to a pipestage when a prior pipestage contains an invalid instruction as indicated by a valid indicator of the instruction to conserve power.

26. A computer including:

a memory; and

a microprocessor, the microprocessor including,

an instruction decoder to decode multiple threads of instructions, the instruction decoder having,

an instruction decode pipeline to decode each input instruction associated with each thread; and

a shadow pipeline operating in parallel with the instruction decode pipeline, the shadow pipeline including, an instruction thread identification pipeline to associate a thread identification with each instruction being decoded in the instruction decode pipeline, and

an instruction valid pipeline to associated a valid indicator with each instruction being decoded in the instruction decode pipeline.

27. A method of eliminating invalid instructions within an instruction decoder comprising:

receiving a thread specific clear instruction indicating a thread ID of instructions to be cleared from a pipeline;

comparing a thread identifier of each instruction within each pipestage of the pipeline to determine if it matches the thread ID of instructions to be cleared from the pipeline;

invalidating a valid bit for each instruction having a thread ID matching the thread ID of instructions to be cleared from the pipeline.

28. The method of claim 17 further comprising:

clocking each pipestage of the pipeline to continue to decode the valid instructions within the pipeline as indicated by their valid bits.

29. A microprocessor including:

an instruction decoder to decode multiple threads of instructions, the instruction decoder having,

an instruction decode pipeline,

the instruction decode pipeline to decode each input instruction associated with each thread, and

the instruction decode pipeline to maintain a thread identification and a valid indicator in parallel with each instruction being decoded in the instruction decode pipeline.

30. The microprocessor of claim 29 wherein the instruction decode pipeline includes a series of registers within a microprocessor integrated circuit.

31. The microprocessor of claim 29 wherein the instruction decoder further has,

a pipeline controller coupled to the instruction decode pipeline, the pipeline controller to control the clocking of each pipestage of the instruction decode pipeline.

32. The microprocessor of claim 31 wherein a thread specific clear is received and the pipeline controller invalidates only those instructions in each stalled pipestage of the instruction decode pipeline having a thread identification the same as the thread identification of the thread specific clear.

33. The microprocessor of claim 31 wherein a thread specific clear is received and the pipeline controller invalidates only those instructions input into each clocked pipestage of the instruction decode pipeline when a pipestage prior to each clocked pipestage has a thread identification the same as the thread identification of the thread specific clear.

34. The microprocessor of claim 31 wherein a stall associated with a thread identification is received and the pipeline controller determines that the thread identification associated with the stall matches the thread identification of the valid instruction in the next to last pipestage of the instruction decode pipeline and the pipeline controller stalls the next to last pipestage by stopping the clock to the next to last pipestage for the next cycle to hold the valid instruction contained therein.

35. The microprocessor of claim 38 wherein the pipeline controller further determines that a valid instruction is contained within a first pipestage of the instruction decode pipeline other than the next to last pipestage and a valid instruction is contained within a second pipestage immediately subsequent to the first pipestage and the pipeline controller stalls the first pipestage by stopping the clock to the first pipestage for the next cycle to hold the valid instruction contained therein.

36. The microprocessor of claim 31 wherein the pipe controller further determines that an invalid instruction is contained within a first pipestage immediately prior to a second pipestage in the instruction decode pipeline and the pipe controller powers down the second pipestage by stopping the clock to the second pipestage until a valid instruction is contained within the first pipestage to conserve power.